Sociedad Colombiana de Matemáticas
XV Congreso Nacional de Matemáticas
2005 **Apuntes**

Lecturas Matemáticas
Volumen Especial (2006), páginas 161–174

Sobre dos teoremas de incompletez de Chaitin

Carlos Mario Parra & Johany A. Suárez
Universidad Nacional de Colombia, Medellín, Colombia

Abstract. Inspired by the so called Hilbert Program and the necessity to clarify the methods of mathematical reasoning , the research on the foundations of mathematics led to the discovery of the Incompleteness phenomenon. Using the notion of algorithmic complexity of an object it is possible to obtain a new array of results that exhibit the ubiquity and relevance of such a phenomenon. In this paper we sketch the basic notions of Information and Randomness and propose another definition of the complexity of a formal system, which allows us to give alternative proofs of two classical incompleteness theorems by G. Chaitin.

Key words and phrases. Algorithmic Complexity, Incompleteness, Chaitin computer.

2000 AMS Mathematics Subject Classification. Primary: 68Q30. Secondary: 03D10.

Resumen. Impulsadas por el Programa de Hilbert y la necesidad de clarificar los métodos del razonamiento matemático, las investigaciones sobre los fundamentos de la matemática condujeron al descubrimiento del fenómeno de incompletez. Mediante la noción de complejidad algorítmica de un objeto, es posible obtener una nueva gama de resultados que exhiben la generalidad y relevancia de dicho fenómeno. En este artículo bosquejamos las nociones básicas de aleatoriedad e información y proponemos otra definición de la complejidad de un sistema formal, lo cual nos permite ofrecer pruebas alternativas de dos resultados clásicos de incompletez, debidos a G. Chaitin.

1. Introducción

Durante los años sesenta, Kolmogorov propuso un enfoque algorítmico para la teoría de la información de Shannon. Su principal motivación era proporcionar una definición aceptable de *sucesión aleatoria*, mediante la idea

de *complejidad algorítmica*. Intuitivamente, la complejidad (cantidad de información) de un objeto finito es la longitud de la entrada más corta que al ingresarse, sin datos adicionales, a una máquina de Turing universal genera el objeto. Así, las sucesiones aletorias son aquellas que poseen segmentos iniciales de máxima complejidad. La noción de complejidad algorítmica le permitió a Chaitin presentar versiones cuantitativas del teorema de Gödel. En términos informales:

> Un sistema formal axiomático cuya complejidad sea N no puede derivar un teorema que afirme que un objeto específico posee complejidad substancialmente mayor que N.

En este artículo, desarrollamos las nociones básicas de información y aleatoriedad suficientes para enunciar formalmente dos teoremas de incompletez de Chaitin. Además, proponemos una definición diferente de complejidad de una teoría, lo que nos permitirá dar pruebas alternativas de dichos teoremas. En lo que sigue seguiremos el tratamiento y la notación de [1].

2. **Preliminares**

Dado un *alfabeto* $A = \{a_1, \ldots, a_Q\}$, con $Q \geq 2$, representamos por A^* el conjunto infinito de todas las *cadenas* x sobre A, incluyendo la *cadena vacía* λ. Para $x \in A^*$, $|x|$ denota la *longitud* de x ($|\lambda| = 0$). La *concatenación* de $x, y \in A^*$ se denota por xy. A^ω denota el conjunto de todas las *sucesiones* infinitas $\mathbf{x} = x_1 x_2 \ldots x_n \ldots$, con $x_i \in A$. El *segmento inicial* $\mathbf{x}(n)$ es el prefijo de $\mathbf{x}$ de longitud $n > 0$; i.e., $\mathbf{x}(n) = x_1 x_2 \ldots x_n \in A^*$.

En cierto momento, necesitaremos considerar los elementos del alfabeto A como los dígitos en base Q. Para enfatizar este hecho denotaremos A por A_Q y escribiremos $A_Q = \{0, 1, \ldots, Q-1\}$. De esta forma, a cada sucesión $\mathbf{x} \in A_Q^\omega$ le asociamos su *valor* en $[0, 1]$ por

$$\nu_Q(\mathbf{x}) = 0.\mathbf{x} = \sum\nolimits_{n=1}^{\infty} x_n \cdot Q^{-n}.$$

Análogamente, definimos el valor $\nu_Q(x) = 0.x$ de la cadena $x \in A_Q^*$. Asimismo, cada $\alpha \in [0, 1]$ se identifica de forma unívoca con la sucesión de dígitos de su expansión en la base Q. En el caso $\alpha = k \cdot Q^{-n}$, tomamos la expansión que no termina con infinitos ceros.

Cualquier orden total entre los símbolos de A induce el denominado *orden lexicográfico* en A^*. Denotamos por $\mathsf{string}(n)$ la enésima cadena de acuerdo al orden lexicográfico. De esta forma, obtene-mos una función biyectiva computable $\mathsf{string} : \mathbb{N} \to A^*$. Además $|\mathsf{string}(n)| = \lfloor \log_Q(n(Q-1)+1) \rfloor$.

Una cadena x es *prefijo* de otra cadena y (lo que se denota $x <_p y$), si $y = xz$, para algún $z \in A^*$. Un conjunto $S \subseteq A^*$ se dice *libre de prefijos*, si dados $x, y \in S$ se tiene que $x <_p y$, implica $x = y$. El siguiente es un resultado clásico de la teoría de la información.

Teorema 2.1 (KRAFT). *Si $S \subseteq A^*$ es libre de prefijos, entonces*

$$\sum\nolimits_{s\in S} Q^{-|s|} \leq 1.$$

Demostración. Para $S = \{s_1, \ldots, s_n\}$ finito, hacemos r_k igual al número de elementos de S de longitud k. Sea $m = \max\{|s| \mid s \in S\}$. Es claro que $r_k = 0$ si $k > m$. Como S es libre de prefijos se cumple que

$$\begin{aligned} r_1 &\leq Q \\ r_2 &\leq (Q - r_1)\, Q = Q^2 - r_1 Q \\ r_3 &\leq ((Q - r_1)\, Q - r_2)\, Q = Q^3 - r_1 Q^2 - r_2 Q \\ &\vdots \\ r_m &\leq Q^m - r_1 Q^{m-1} - \cdots - r_{m-1} Q \end{aligned}$$

Dividiendo la última desigualdad por Q^m se obtiene

$$\sum\nolimits_{s\in S} Q^{-|s|} = \sum\nolimits_{i=1}^{m} r_i Q^{-i} \leq 1.$$

Cuando S es infinito, se tiene que

$$\sum\nolimits_{s\in S} Q^{-|s|} = \sup_T \sum\nolimits_{t\in T} \leq 1,$$

donde el supremo se toma sobre todos los subconjuntos finitos T de S. ☑

Con el fin de representar los números naturales por medio de un conjunto libre de prefijos, introducimos la función $bin : \mathbb{N}^+ \to \{0,1\}^*$, donde $bin(n)$ es la única cadena binaria tal que $1bin(n)$ es igual a la expansión binaria de n. Ahora, con cada $x \in \{0,1\}^*$ construimos una nueva cadena $\overline{x}$ insertando un 0 antés de cada bit de x, y al final adicionando un 1. Por ejemplo, $\overline{1011} = 01\ 00\ 01\ 01\ 1$. Por último, hacemos $d(x) = \overline{bin(|x|)}x$, para todo $x \in \{0,1\}^*$. $d(x)$ se denomina la *versión autolimitante binaria* de x. Se tiene que $S = \left\{d(x) : x \in \{0,1\}^*\right\}$ es libre de prefijos y que toda cadena $x \in \{0,1\}^*$ se puede representar en de S usando $|d(x)| = |x| + 2\log_2 |x| + 1$ bits. Por tanto, todo $n \in \mathbb{N}^+$ tiene una representación en S de longitud $\log_2 n + 2\log_2 (\log_2 n) + 1$ bits. Además, reemplazando 0 por a_1 y 1 por a_2 podemos considerar que la funcion bin toma valores en el alfabeto arbitrario A. De este modo, el conjunto $\{d(x) : x \in A^*\} \subseteq A^*$ es libre de prefijos, donde $d(x) = \overline{bin(|x|)}x$ es la *versión autolimitante* de $x \in A^*$.

Una *función parcial* de X en Y es una función definida en un subconjunto de X, lo que denotaremos por $\varphi : X \xrightarrow{o} Y$. En caso de que $dom(\varphi) = X$, decimos que φ es *total* e indicamos esto escribiendo $\varphi : X \to Y$. Cuando $x \in dom(\varphi)$, escribimos $\varphi(x) \neq \infty$; en caso contrario, escribiremos $\varphi(x) = \infty$.

Una función parcial $\varphi : A^* \xrightarrow{o} A^*$ se llama *parcial recursiva* (abreviado *p.r.*) si existe una función parcial recursiva $f : \mathbb{N} \xrightarrow{o} \mathbb{N}$ tal que

$$\varphi(x) = \mathsf{string}\left(f\left(\mathsf{string}^{-1}(x)\right)\right), \qquad \forall x \in A^*.$$

Análogamente, definimos las funciones *recursivas* en A^*. Un conjunto $X \subseteq A^*$ es *recursivamente enumerable* (abreviado *r.e.*) si es vacío o es el rango de una función p.r. Equivalentemente, X es r.e. si es el dominio de una función p.r. Decimos que X es *recursivo* si X y $A \setminus X$ son r.e.

Como es usual en computabilidad y lógica, haremos uso de *gödelizaciones* del conjunto de expresiones del lenguaje $\mathcal{L}$ de alguna teoría formal $\mathcal{T}$ o del conjunto de todas la máquinas de Turing. En este último caso, $\varphi_e^{(n)}$ denota la función p.r. de n variables computada por la máquina de Turing con número de Gödel $e \geq 1$.

3. **Complejidad y aleatoriedad**

Para formalizar adecuadamente la noción de contenido de información, visualizamos un computador como una función parcial recursiva que recibe programa + datos como entradas y que luego puede imprimir otra cadena como salida. Esto es, un *computador* es una función p.r. $\varphi : A^* \times A^* \xrightarrow{o} A^*$. Un *algoritmo libre de prefijos* es una función p.r. $\varphi : A^* \xrightarrow{o} A^*$, cuyo dominio es libre de prefijos. Podemos imaginar un algoritmo libre de prefijos como una función que es computable por una *máquina de Turing autolimitante*. En este tipo de máquina la cabeza lectora está restringida a leer la entrada en una dirección sólo si los blancos no se permiten como "marcadores de fin". Decimos que una máquina autolimitante M ejecuta una computación exitosa en la entrada p si M para cuando la cabeza lectora escanea el último bit de p. Se sigue entonces que, a diferencia de los algoritmos ordinarios, los algoritmos libres de prefijos son autolimitantes en el siguiente sentido: *la extensión de un programa válido no es un programa válido.*

La anterior discusión motiva el reemplazo de la noción clásica de complejidad, debida a Kolmogorov, por la versión autolimitante. Definimos un *computador de Chaitin* como un computador $C : A^* \times A^* \xrightarrow{o} A^*$ tal que el dominio de C_v es libre de prefijos, para todo $v \in A^*$. Aquí, $C_v : A^* \xrightarrow{o} A^*$ se define por

$$C_v(u) = C(u, v), \qquad \forall u \in A^*.$$

Esto es, si $C(u, v) \neq \infty$ y $y <_p u$, entonces $C(y, v) \neq \infty$ implica que $y = u$.

Un computador de Chaitin U es *universal*, si para todo computador de Chaitin C, existe una constante c (que depende de ambos computadores) para la cual se cumple que si $C(x, v) \neq \infty$, entonces existe una cadena x' tal que $U(x', v) = C(x, v)$ y $|x'| \leq |x| + c$. La constante c representa el costo de simular el computador C en el computador universal.

Un resultado clásico de la teoría de la computabilidad establece la existencia de máquinas universales que simulan efectivamente cualquier máquina de Turing usual. Análogamente, existe una función universal que sólo simula computadores de Chaitin (véase [7] o [10]).

Lema 3.1. *Existe una función p.r.* $F : \mathbb{N}^+ \times A^* \times A^* \overset{o}{\to} A^*$ *tal que si* $\varphi_n : A^* \times A^* \overset{o}{\to} A^*$ *es un computador de Chaitin y* $\varphi_n(u,v) \neq \infty$, *entonces*

$$F(n,u,v) = \varphi_n(u,v).$$

Ahora, si tomamos la función universal $F : \mathbb{N}^+ \times A^* \times A^* \overset{o}{\to} A^*$ del lema 3.1 y definimos el computador de Chaitin $U : A^* \times A^* \overset{o}{\to} A^*$ por

$$U\left(a_1^i a_2 u, v\right) = F(i,u,v), \qquad \text{donde } a_1, a_2 \in A,$$

obtenemos un computador de universal.

En adelante, fijaremos un computador universal de Chaitin U y lo usaremos para medir la complejidad de longitud de programa. De forma análoga a la complejidad de Kolmogorov, definimos la *complejidad absoluta autolimitante de Chaitin* (complejidad de Chaitin) asociada con el computador de Chaitin C como la función parcial $H_C : A^* \overset{o}{\to} \mathbb{N}$, dada por

$$H_C(x) = \begin{cases} \infty & \text{si no existe } u \text{ tal que } C(u,\lambda) = x, \\ \min\{|u| : C(u,\lambda) = x\} & \text{en otro caso.} \end{cases}$$

Cuando $C = U$, hacemos $H(x) = H_U(x)$.

Como $\lambda \notin dom(U_\lambda)$ y U_λ es sobreyectiva, entonces para todo $x \in A^*$ existe al menos un programa $u \neq \lambda$ tal que $U(u,\lambda) = x$. De acuerdo al orden lexicográfico de A^*, denotamos por x^* la menor cadena del conjunto no vacío $\{u \in A^* : U(u,\lambda) = x\}$; x^* se suele denominar el *programa canónico* de x. Es obvio que $H(x) = |x^*|$. A partir de la definición de complejidad es inmediato el siguiente resultado.

Teorema 3.1 (Teorema de Invarianza)**.** *Para todo computador de Chaitin* C *se tiene que*

$$H(x) \leq H_C(x) + O(1).$$

El siguiente lema nos será de utilidad más adelante.

Lema 3.2. *Para todo* $n \in \mathbb{N}^+$, $H(\mathsf{string}(n)) = O(\log_Q n)$.

Demostración. Consideremos el computador de Chaitin $C(d(x),\lambda) = x$, donde $d(x)$ es la versión autolimitante de la cadena $x \in A^*$. Dado que $|d(x)| = |x| + 2\log_2|x| + 1$, existe una constante positiva c tal que para todo $x \neq \lambda$:

$$H(x) \leq H_C(x) + c \leq |x| + 2\log_2|x| + c$$

En particular,

$$\begin{aligned} H(\mathsf{string}(n)) &\leq |\mathsf{string}(n)| + 2\log_2|\mathsf{string}(n)| + c \\ &= \left\lfloor \log_Q(n(Q-1)+1) \right\rfloor + 2\log_2\left\lfloor \log_Q(n(Q-1)+1) \right\rfloor + c. \end{aligned}$$

Por tanto, $H(\mathsf{string}(n)) = O(\log_Q n)$. ☑

La gran ventaja de trabajar con máquinas autolimitantes es que permiten generalizar la *desigualdad de Kraft* (teorema 2.1) lo cual proporciona criterios

para el diseño de computadores de Chaitin y consecuentemente la obtención de cotas superiores. Dicha generalización se conoce como el *teorema de Kraft–Chaitin*, cuyo enunciado y demostración se pueden consultar en [1] o [10].

Uno de los grandes logros de la teoría algorítmica de la información es mostrar que la noción de sucesión *incomprimible*, propuesta por Kolmogorov, coincide con las nociones de aleatoriedad de Martin-Löf y de Solovay, si adoptamos la complejidad autolimitante de Chaitin (véase teorema 6.35 en [1]). De acuerdo a la anterior discusión introducimos la siguiente definición.

Definición 3.2 (Chaitin–Schnorr)**.** Una sucesión $\mathbf{x} \in A^{\omega}$ es *aleatoria* si y sólo si existe una constante $c > 0$ tal que para todo $n \geq 1$

$$H\left(\mathbf{x}\left(n\right)\right) \geq n - c.$$

De hecho, se puede probar que la complejidad de los segmentos iniciales $\mathbf{x}(n)$ de una sucesión aleatoria $\mathbf{x}$ se aleja arbitrariamente de su longitud n, a medida que n tiende a infinito (véase el capítulo 6 en [1]).

Teorema 3.3 (Chaitin)**.** $\mathbf{x}$ *es aleatoria si y sólo si* $\lim_{n\to\infty} H\left(\mathbf{x}(n)\right) - n = \infty$.

Intuitivamente, la más insignificante posibilidad de computar una fracción infinita de una sucesión hace que ésta no sea aleatoria. Dicha intuición se formaliza en la siguiente proposición que necesitaremos en la siguiente sección (para una prueba, véase el teorema 6.41 en [1]).

Proposición 3.4. Sea $\mathbf{x} \in A^{\omega}$. Si existe una sucesión estrictamente creciente de naturales $i(k), k \geq 1$ tal que el conjunto $\left\{\left(i(k), x_{i(k)}\right) \mid k \geq 1\right\}$ es recursivo, entonces $\mathbf{x}$ no es aletoria.

4. **Incompletez via información**

En los años setenta, Chaitin desarrolló las primeras generalizaciones de la incompletez, via información, basándose en la idea de definir la complejidad de un sistema formal axiomático $\mathcal{T}$ como la longitud del menor programa que *genera el algoritmo que verifica pruebas* dentro de la teoría (*proof-checking algorithms*). Pero este enfoque es bastante restrictivo pues sólo se aplica a *teorías finitamente axiomatizables* (véase [2] y [3]). En los años noventa, Chaitin consideró la posibilidad de medir la complejidad algorítmica de computar conjuntos infinitos. Para ello, introdujo una nueva noción: los computadores de enumeración (*e-computers*), que corresponden a una familia de máquinas de Turing autolimitantes que pueden imprimir listas infinitas de cadenas en su cinta de salida. De este modo, se modifica la definición de la complejidad de $\mathcal{T}$ como la longitud del menor programa que, cuando se le ingresa a un e-computador universal, *genera todos los teoremas* de $\mathcal{T}$. Esto le permite a Chaitin ampliar las variantes informáticas del teorema de Incompletez a *teorías con infinitos axiomas*. Sin embargo, las cotas que se obtienen involucran constantes que *dependen* del sistema considerado.

Ahora proponemos otra definición de complejidad de un sistema formal, la cual nos permitirá modificar las pruebas dadas en [4]. Así, logramos una extensión de los resultados clásicos de Chaitin, de modo que éstos no sólo se aplican a teorías arbitrarias, sino que además las constantes involucradas en las cotas no dependen del sistema formal considerado.

Sea $\mathcal{T}$ una teoría formal axiomatizable cuyo lenguaje $\mathcal{L}$ incluye al lenguaje de la *Aritmética*. Como $\mathcal{T}$ es axiomatizable existe al menos una función primitiva recursiva φ cuyo rango es el conjunto de números de Gödel del conjunto de teoremas de $\mathcal{T}$. Ahora fijamos una gödelización $\mathcal{G}$ de las funciones primitivas recursivas y, sin pérdida de generalidad, suponemos que todo $n \in \mathbb{N}$ "codifica" una función primitiva recursiva que denotamos por φ_n. Definimos la *complejidad algorítmica de* la teoría axiomatizable $\mathcal{T}$ como

$$H(\mathcal{T}) = \min\ H(\mathsf{string}(n)),$$

donde n varía sobre todos los números de Gödel[1] de las funciones primitivas recursivas φ_n que generan a $\mathcal{T}$.

Es claro que hay *finitos* naturales n tales que $H(\mathsf{string}(n))$ es mínimo y φ_n enumera los teoremas de $\mathcal{T}$. En otras palabras, hay finitos $u \in A^*$ tal que $H(\mathcal{T}) = H(u) = |u^*|$. Tomemos

$$w = \min\{u \in A^* \mid H(\mathcal{T}) = H(u)\}$$

según el orden lexicográfico. Así, existe un único $m \in \mathbb{N}$ tal que $\mathsf{string}(m) = w$ y si hacemos $v = w^*$ obtenemos que φ_m enumera los teoremas de $\mathcal{T}$ y además $H(\mathcal{T}) = |w^*| = |v|$. Por brevedad, denotamos φ_m por $\varphi_{\mathcal{T}}$. Las anteriores consideraciones nos permiten obtener la primera versión del teorema de Incompletez de Chaitin.

Teorema 4.1. *Existe una constante* c *tal que si* $\mathcal{T}$ *es una teoría axiomatizable y sólida, entonces* $\vdash_{\mathcal{T}} H(s) > n$, *para* $s \in A^*$, *sólo si* $n < H(\mathcal{T}) + \mathsf{c}$.

Demostración. Consideremos el computador de Chaitin C que, dado $w \in A^*$, ejecuta el siguiente procedimiento:

1. `genere el conjunto` $W = \{x \mid x <_p w \wedge x \neq w\}$.
2. `genere sistemáticamente (dovetailing) las computaciones` $U(x, \lambda)$, `con` $x \in W$, `hasta que halle una computación que pare`.
3. `haga` $u := \min\{x \in W : U(x, \lambda) \neq \infty\}$ `y` $k := \mathsf{string}^{-1}(U(u, \lambda))$.
4. `compute` $U(v, \lambda)$, `donde` $uv = w$.
5. `si` $U(v, \lambda) \neq \infty$, `entonces haga` $n = \mathsf{string}^{-1}(U(v, \lambda))$ `y de` n `recupere la función` φ_n.
6. `genere el rango de` φ_n `hasta que halle una sentencia de la forma` $H(s) > |v| + k$ `e imprima la correspondiente cadena` $s \in A^*$.

[1] Debemos distinguir entre la gödelización dada a las expresiones del lenguaje $\mathcal{L}$ y la dada a las funciones primitivas recursivas φ_n.

Es decir, si $U(u,\lambda) = \mathsf{string}(k)$ y $U(v,\lambda) = \mathsf{string}(\mathcal{G}(\varphi_{\mathcal{S}}))$, donde $\mathcal{S}$ es una teoría axiomatizable cualesquiera, entonces $C(uv,\lambda)$ es la primera cadena $s \in A^*$ para la cual $\mathcal{S}$ prueba que su complejidad es mayor que $|v|+k$. Como $dom(U_\lambda)$ es libre de prefijos, $dom(C_\lambda)$ también es libre de prefijos. Por el Teorema de Invarianza, existe una constante d (que sólo depende de U y C) tal que

$$H(s) \leq H_C(s) + d.$$

Ahora razonemos por el absurdo y supongamos que para toda constante c, existe una teoría axioma-tizable y sólida $\mathcal{T}$ tal que $\vdash_{\mathcal{T}} H(s) > H(\mathcal{T}) + \mathsf{c}$, para alguna cadena $s \in A^*$.

Como $H(\mathsf{string}(k)) + d = O(\log_Q k)$ (véase el lema 3.2), podemos tomar $\mathsf{c} \geq 1$ que **viole** la desigualdad

$$k < H(\mathsf{string}(k)) + d.$$

Entre las entradas admisibles para C, podemos hallar la concatenación de los programas canónicos para $\mathsf{string}(\mathsf{c})$ y para $\mathsf{string}(\mathcal{G}(\varphi_{\mathcal{T}}))$. Es decir, si $u = \mathsf{string}(\mathsf{c})^*$ y $v = \mathsf{string}(\mathcal{G}(\varphi_{\mathcal{T}}))^*$, entonces $C(uv,\lambda) \neq \infty$.
Si $s = C(uv,\lambda)$, se cumple que

$$H_C(s) \leq |u| + |v| = H(\mathsf{string}(\mathsf{c})) + H(\mathcal{T}).$$

Por el paso 6, $\vdash_{\mathcal{T}} H(s) > |v| + \mathsf{c}$. Por tanto

$$\mathsf{c} + H(\mathcal{T}) = \mathsf{c} + |v| < H(s) \leq H(\mathsf{string}(\mathsf{c})) + H(\mathcal{T}) + d,$$

lo cual contradice la elección de c. ☑

En 1974, Chaitin introdujo la *probabilidad de parada (Omega)*

$$\Omega = \sum\nolimits_{u \in \mathrm{dom}(\mathrm{U}_\lambda)} Q^{-|u|}.$$

Como $\mathrm{dom}(\mathrm{U}_\lambda)$ es libre de prefijos, por la desigualdad de Kraft, $0 \leq \Omega \leq 1$. Intuitivamente, Ω representa la "probabilidad" de que un computador de Chaitin universal U se detenga, si el programa $u \in A^*$ se toma al azar y no se ingresan datos adicionales. Es importante observar que Ω depende del computador universal U (equivalentemente, depende de la gödelización de las máquinas de Turing autolimitantes) que se halla fijado. Así, lo que tenemos realmente es una familia de números Ω, no una constante absoluta como π o e. Pero **todas** estas "probabilidades" comparten propiedades muy interesantes, algunas de las cuales discutiremos en esta sección. Por ahora fijemos un computador autolimitante universal U y estudiemos el número Ω asociado a él.

Tomemos una función recursiva inyectiva $f : \mathbb{N}^+ \to A^*$ tal que $f(\mathbb{N}^+) = dom(U_\lambda)$ [2] y

$$\omega_k = \sum\nolimits_{i=1}^{k} Q^{-|f(i)|}.$$

[2]Generando sistemáticamente todas las computaciones $U(u,\lambda)$, se puede mostrar que $dom(U_\lambda)$ es r.e.

Es claro que $(\omega_k)_{k\geq 0}$ es una sucesión creciente de números racionales que converge a Ω. Los números reales que, como Ω, se pueden aproximar desde abajo por una sucesión creciente y recursiva de números racionales se llaman *reales recursivamente enumerables* (véase la sección 7.4 en [1]). Denotemos por

$$0.\Omega_1\Omega_2\cdots\Omega_n\cdots\in(0,1)$$

la expansión infinita[3] de Ω en la base Q y por $\Omega(n)$ la aproximación racional

$$0.\Omega_1\Omega_2\ldots\Omega_n$$

Inicialmente veamos la forma en que Ω codifica soluciones al problema de la parada.

Proposición 4.2. Dado $\Omega(n)$ podemos decidir si $U(u,\lambda)\neq\infty$, para todo $u\in A^*$ tal que $|u|\leq n$.

Demostración. Dado $u\in A^*$ con $|u|\leq n$, generemos el conjunto de programas

$$W_k:=\{f(1),\ldots,f(k)\}\subset dom(U_\lambda)$$

hasta que hallemos que $\Omega(n)<\omega_k$. Ahora si $U(u,\lambda)\neq\infty$ y $u\notin W_k$, entonces

$$\Omega(n)\leq\Omega\leq\Omega(n)+Q^{-n}<\omega_k+Q^{-|u|}\leq\sum\nolimits_{i=1}^{\infty}Q^{-|f(i)|}=\Omega,$$

absurdo. Por tanto, $U(u,\lambda)\neq\infty$ *si y sólo si* $u\in W_k$. ☑

Como consecuencia de la proposición 4.2, el conocimiento de $\Omega(n)$ es suficiente para determinar si $H(x)\leq n$ para toda cadena $x\in A^*$. Conocer los primeros $10,000$ dígitos de Ω nos permite decidir sobre el detenimiento de todos los programas cuya longitud sea menor que $10,000$, lo cual incluye programas que buscan sistemáticamente contraejemplos para la *Conjetura de Goldbach*, la *Hipótesis de Riemann* y muchas otras conjeturas en matemáticas que se pueden refutar mediante un solo contraejemplo finito. Pero aún si uno posee $\Omega(10,000)$, hacer uso de la información que codifica requiere un gasto de tiempo que está más alla de cualquier aplicación práctica. De hecho, el tiempo $t(n)$ que toma hallar todos los programas que paran con longitud menor que n a partir de $\Omega(n)$ crece más rápido que cualquier función recursiva ([7]).

Recordemos que los dígitos $\{0,\ldots,Q-1\}$ se pueden asociar a los elementos de $A=\{a_0,\ldots,a_{Q-1}\}$. Entonces $\Omega_1\Omega_2\ldots\Omega_n\ldots$ es una sucesión de A^ω que denotaremos por $\mathbf{r}_Q(\Omega)$.

Teorema 4.3 (Chaitin)**.** *La sucesión* $\mathbf{r}_Q(\Omega)\in A^\omega$ *es aleatoria.*

Demostración. Definamos un computador de Chaitin C que, al leer el programa $u\in A^*$, sigue el siguiente procedimiento:

[3]Como aún no sabemos si Ω es un número irracional, de tener dos expansiones, escogemos la que no termine en infinitos ceros.

1. `compute` $U(u,\lambda)$.
2. `si` $U(u,\lambda) \neq \infty$, `entonces haga` $x := U(u,\lambda)$.
3. `halle el menor` k `tal que` $\omega_k \geq \nu_Q(x) = 0.x$.
4. `imprima la menor cadena que no pertenezca a` $\{U(f(1),\lambda),\ldots,U(f(k),\lambda)\}$.

Notemos que $C(u,\lambda) = \infty$, si $U(u,\lambda) = \infty$ o si no existe un k como el pedido en el paso 3. Si $C(u,\lambda) \neq \infty$ y v es otro programa con $U(u,\lambda) = U(v,\lambda)$, entonces $C(u,\lambda) = C(v,\lambda)$. Aplicando esto a un programa arbitrario $u \in dom(C_\lambda)$ y al programa canónico $v = (U(u,\lambda))^*$ de $U(u,\lambda)$, tenemos que

$$H_C(C(u,\lambda)) \leq |v| = H(U(u,\lambda)).$$

Por el Teorema de Invarianza, existe una constante positiva c tal que para todo $u \in dom(C_\lambda)$:

$$H(C(u,\lambda)) \leq H_C(C(u,\lambda)) + c \leq H(U(u,\lambda)) + c. \tag{4.1}$$

Ahora, sea n fijo y supongamos que u es un programa tal que

$$U(u,\lambda) = \mathbf{r}_Q(\Omega)(n) = \Omega_1\Omega_2\ldots\Omega_n \in A^*.$$

Entonces $C(u,\lambda) \neq \infty$. Sea k el menor número que cumple $\omega_k \geq 0.\Omega_1\Omega_2\ldots\Omega_n$ (computado en el paso 3 y cuya existencia es asegurada por las definiciones dadas). Por tanto,

$$0.\Omega_1\Omega_2\ldots\Omega_n \leq \omega_k < \omega_k + \sum\nolimits_{i=k+1}^{\infty} Q^{-|f(i)|} = \Omega \leq 0.\Omega_1\Omega_2\ldots\Omega_n + Q^{-n}.$$

Luego

$$\sum\nolimits_{i=k+1}^{\infty} Q^{-|f(i)|} \leq Q^{-n}.$$

Esto implica que $|f(i)| > n$, para todo $i \geq k{+}1$. Por el paso 4 de la construcción de C concluimos que $n < H(C(u,\lambda))$ y usando (4.1), obtenemos que

$$n < H(U(u,\lambda)) + c = H(\Omega_1\Omega_2\ldots\Omega_n) + c.$$

Por la definición de Chaitin–Schnorr, la sucesión $\mathbf{r}_Q(\Omega)$ es aleatoria. ☑

Como consecuencias del teorema 4.3, Ω es un número real *aleatorio* (independientemente de la base elegida [9]), *no computable* y *Borel normal* (en cualquier base). Como todo número algebraico es computable, tenemos que Ω es *trascendente*. Calude, Dineen y Shu han calculado los primeros 64 bits exactos de cierto Ω_U :

0000001000 0001000001 1000100001 1010001111 1100101110 1110100001 0000

(véase la sección 8.7 en [1]). Sin embargo, hay limitaciones intrínsecas relacionadas con el número de dígitos de Ω que cualquier sistema formal pueda hallar.

Decimos que la teoría formal $\mathcal{T}$ *determina la posición y el valor de un dígito de* Ω si existen n, i tales que la sentencia

$$\mathcal{A}(n,i) \equiv \text{“el } \textit{enésimo} \text{ dígito de } \Omega \text{ es } a_i\text{”} \equiv \text{“ } \Omega_n = a_i \text{ ”}$$

es un teorema de $\mathcal{T}$.

Teorema 4.4. *Sea $\mathcal{T}$ una teoría axiomatizable y sólida, entonces $\mathcal{T}$ determina la posición y el valor de a lo sumo finitos dígitos (dispersos) de Ω.*

Demostración. Supongamos que $\mathcal{T}$ puede determinar infinitas posiciones y valores de Ω. Es decir, existe un conjunto infinito de posiciones y valores de Ω que es r.e. A partir de dicho conjunto, podemos obtener una función creciente $i : \mathbb{N}^+ \to \mathbb{N}^+$ tal que el conjunto

$$\left\{ \left(i\left(k\right), \Omega_{i(k)} \right) \mid k \geq 1 \right\}$$

es recursivo. Por la proposición 3.4, la sucesión $\mathbf{r}_Q\left(\Omega\right)$ no sería aleatoria; absurdo. ☑

De hecho, podemos probar la existencia de una cota para el número de dígitos dispersos de Ω que $\mathcal{T}$ puede determinar. Dicha cota involucra la complejidad $H\left(\mathcal{T}\right)$, junto con una constante independiente de la teoría. Así, obtenemos la segunda versión del teorema de Incompletez de Chaitin.

Teorema 4.5. *Existe una constante c tal que si $\mathcal{T}$ es una teoría axiomatizable y sólida, entonces $\mathcal{T}$ determina las posiciones y valores de a lo sumo $H\left(\mathcal{T}\right) + \mathsf{c}$ dígitos (dispersos) de Ω.*

Demostración. Consideremos un computador de Chaitin C similar al construido en la prueba de la primera versión del teorema de Incompletez. Dado el programa

$$w = \overbrace{a_1 a_1 \cdots a_1 a_1 a_2}^{k \text{ dígitos}} uv,$$

C lee el primer a_1 y continua hasta que halla el primer a_2. La longitud de la cadena leída es k. Luego, simula $U\left(u, \lambda\right) = \mathsf{string}\left(e\right)$. Una vez decodificado e, C enumera el rango de φ_e hasta que determina $|u| + 2k$ dígitos dispersos de Ω. A continuación, C busca la posición n, entre los dígitos generados de Ω, que esté más lejos del punto decimal. En otras palabras, C halla el mayor n entre los primeros $|u| + 2k$ teoremas de la forma $\mathcal{A}\left(n, i\right)$ enumerados por la simulación de φ_e. Ahora, consideremos la cadena $\mathbf{r}_Q\left(\Omega\right)\left(n\right)$ de los primeros n dígitos de Ω en base Q :

$$\mathbf{r}_Q\left(\Omega\right)\left(n\right) = \beta_1 \beta_2 \cdots \beta_n.$$

Hasta este momento de la computación, C ha determinado la posición y el valor de $|u| + 2k$ símbolos de la cadena $\mathbf{r}_Q\left(\Omega\right)\left(n\right) \in A^n$. Los símbolos restantes son proporcionados por el resto v del programa w como sigue: si $|v| = n - (|u| + 2k)$, C imprime la cadena $x \in A^*$ de longitud n que se construye con los dígitos de Ω determinados por la simulación de φ_e y en la cual las posiciones faltantes se llenan con los símbolos que conforman a v en el orden en que aparecen; después C se detiene. De lo contrario, C entra en un ciclo infinito. Supongamos que

$C(w,\lambda) \neq \infty$ y que $w <_p w'$ y $w \neq w'$. Entonces

$$w' = \overbrace{a_1 a_1 \cdots a_1 a_1 a_2}^{k \text{ dígitos}} uv'.$$

Pero $|v'| > n - (|u| + 2k)$, por lo que $C(w', \lambda) = \infty$. De este modo, $dom(C_\lambda)$ es libre de prefijos.

Ahora, por el Teorema de Invarianza, existe una constante c_0 tal que

$$H(x) \leq H_C(x) + c_0.$$

Como la sucesión $\mathbf{r}_Q(\Omega) \in A^\omega$ es aleatoria, existe una constante c_1 tal que para todo n :

$$n - c_1 < H(\mathbf{r}_Q(\Omega)(n)).$$

Supongamos, por el absurdo, que para toda constante c existe una teoría axiomatizable y sólida $\mathcal{T}$ que determina al menos $H(\mathcal{T}) + 2\mathsf{c}$ dígitos de Ω. Si tomamos $\mathsf{c} = c_0 + c_1$, notamos que entre los programas válidos para C está

$$w = \overbrace{a_1 a_1 \cdots a_1 a_1 a_2}^{\mathsf{c} \text{ dígitos}} uv,$$

donde $u = \mathsf{string}(\mathcal{G}(\varphi_\mathcal{T}))^*$ y v es la cadena con los dígitos faltantes de $\mathbf{r}_Q(\Omega)(n)$. En este caso,

$$C(\overbrace{a_1 a_1 \cdots a_1 a_1 a_2}^{\mathsf{c} \text{ dígitos}} uv, \lambda) = \mathbf{r}_Q(\Omega)(n).$$

Dado que el tamaño del programa w es

$$\left|\overbrace{a_1 a_1 \cdots a_1 a_1 a_2}^{\mathsf{c} \text{ dígitos}}\right| + |u| + |v| = \mathsf{c} + |u| + n - (|u| + 2\mathsf{c}) = n - \mathsf{c},$$

obtenemos que

$$n - c_1 < H(\mathbf{r}_Q(\Omega)(n)) \leq n - \mathsf{c} + c_0.$$

Por tanto $\mathsf{c} < c_0 + c_1$; absurdo. ☑

Según lo anterior, ZFC sólo puede determinar un número finito de dígitos (dispersos) de Ω. Un hecho más sorprendente aún es la creación, por parte de Solovay, de un computador de Chaitin universal para el cual ZFC *no puede decidir ni un solo bit* del Ω asociado a dicho computador (cf. [8] o la sección 8.4 en [1]). Es fácil ver que los dos teoremas anteriores se aplican a cualquier número real aleatorio. Sin embargo, la importancia del número Ω radica en la posibilidad de definirlo en términos matemáticos y en su conexión con el Décimo problema de Hilbert.

Recordemos que una *ecuación Diofantina exponencial* es aquella que se construye mediante adición, multiplicación y exponenciación de variables enteras no negativas, con coeficientes enteros. En 1987, Chaitin construyó una ecuación Diofantina exponencial

$$P(n, x_1, x_2, \ldots, x_m) = 0 \tag{4.2}$$

que tiene sólo un número finito de soluciones $x_1, x_2, \ldots, x_m$ **si y sólo si** el *n-ésimo* bit de Ω es 0 (véase [2], [7] o [10]).

Por el teorema 4.4, podemos concluir que cualquier teoría axiomátizable y sólida $\mathcal{T}$ sólo puede decidir si la ecuación (4.2) tiene finitas o infinitas soluciones para **finitos** valores específicos del parametro n.

Estos resultados muestran claramente una conexión entre el fenómeno de incompletez y las ecuaciones diofantinas. Además, han servido de inspiración a Chaitin para afirmar el descubrimiento del azar y el caos en la aritmética. Los puntos de vista de Chaitin gozan de cierta popularidad, como lo testifica la gran cantidad de artículos en revistas de divulgación sobre el tema [6]. Sin embargo, el lógico holandés van Lambalgen considera que los resultados matemáticos de Chaitin no apoyan sus conclusiones filósoficas y que muchas de sus afirmaciones al respecto se deben a un errónea interpretación de los mismos (cf. [11] y [12]).

Agradecimientos: Los autores recibieron apoyo financiero de la Dirección de Investigación, sede Medellín (DIME) de la Universidad Nacional de Colombia (proyecto *Aleatoriedad y Lógica*, código QUIPU: 030802734).

Bibliografía

[1] C. Calude, *Information and Randomness. An Algorithmic Perspective,* Springer-Verlag: Berlin, 2002.
[2] G. Chaitin, *Algorithmic Information Theory*, Cambridge University Press: Cambrigde, 1987.
[3] G. Chaitin, *Information, Randomness and Incompleteness. Papers on Algorithmic Information Theory*, World Scientific: Singapore, 1990.
[4] G. Chaitin, *Information–Theoretic Incompleteness*, World Scientific: Singapore, 1992.
[5] G. Chaitin, *The Unknowable*, Springer-Verlag: Singapore, 1999.
[6] G. Chaitin, *The Limits of Reason*, Scientific American, **294** (2006), 74–81.
[7] M. Li & P.M. Vintányi, *An Introduction to Kolmogorov Complexity and its Applications*, Springer-Verlag: Berlin, 1993.
[8] R. Solovay, *A Version of Ω for which ZFC can not Predict a Single Bit*, `http://www.cs.auckland.ac.nz/CDMTCS//researchreports/104robert.pdf`
[9] L. Staiger, *The Kolmogorov Complexity of Liouville Numbers*, `http://www.cs.auckland.ac.nz/CDMTCS//researchreports/096Staiger.pdf`
[10] J. Suárez, *El Número Omega. Información, Incompletez y Aleatoriedad*, Tesis de Grado, Universidad de Antioquia: Medellín, 2003.
[11] M. van Lambalgen, *Von Mises' definition of random sequence reconsidered*, J. Symbolic Logic, **52** (1987), 725–755.
[12] M. van Lambalgen, *Algorithmic Information Theory*, J. Symbolic Logic, **54** (1989), 1389–1400.

(Recibido en abril de 2006. Aceptado para publicación en julio de 2006)

Departamento of Matemáticas
Universidad Nacional de Colombia

Medellín, Colombia
e-mail: cmparra@unalmed.edu.co
jasuarezr@unalmed.edu.co